

WEB-BUG DENGAN MEMANFAATKAN VARIABLE SERVER PHP UNTUK MENGUMPULKAN INFORMASI AKTIFITAS PENGUNJUNG WEBSITE

Agung Sasongko

Program Studi Manajemen Informatika, AMIK "BSI Pontianak"

Jl. Abdurahman Saleh No.18A, Pontianak, Indonesia

agung.ako@bsi.ac.id

Abstract

Recording traces of web activity current pengunjung be one way to determine the behavior and habits of website visitors. In e-commerce visitor behavior of data will be very useful to know what became of interest from prospective buyers to read what he was looking for content, so it can be considered a decision to provide the products most frequently searched for visitors. Web-bugs at this time to be one way to record a web visitor activity which works unnoticed by them . This research aims to exploit the server variables contained in webserver, PHP programming khususnya the track record in capturing visitor information website. This study uses an experimental research and application development method using the waterfall model. From the results obtained that an image element in HTML can be used to insert code that can trigger beraksinya web-bug on the server which is then used to record the activity of website visitors.

Keyword: *web-bug, web transaction log, e-commerce*

1. PENDAHULUAN

Ragam pemasaran produk saat ini semakin ramai di Internet, banyak kemudahan bagi para konsumen untuk memilih produk yang sesuai dengan keinginan mereka. Adanya teknologi online akan membuka lebar cara komunikasi antara penjual dan pembeli tanpa dibatasi dengan jarak dan waktu. Informasi-informasi yang disampaikan dapat sangat jelas disajikan, dan setiap calon pembeli dapat juga mengetahui penilaian suatu produk melalui komentar maupun *review* dari pembeli yang sudah pernah menggunakan produk tersebut.

Teknologi *e-commerce* yang canggih seperti yang ditawarkan saat ini, bukan serta merta menjanjikan kesuksesan dalam menjalankan bisnis online. Memiliki informasi dalam membidik target pasar yang tepat sangat dibutuhkan. Menjaga kestabilan dan keamanan sistem yang dimiliki berdasarkan informasi aktifitas *client*

harus dilakukan agar bisnis marketing *online* yang dijalankan dapat bertahan.

Pada penelitian ini penulis akan membahas sekilas teknologi yang dapat dimanfaatkan untuk mengumpulkan informasi dari *client* yang nantinya dapat dipergunakan sebagai pertimbangan untuk membantu *marketer online* menentukan strategi bisnis yang dijalankannya berdasarkan informasi dasar yang didapat dari aktifitas *client* saat mengakses *website*.

Ruang lingkup Penelitian ini membahas bagaimana memanfaatkan *variable server* pada *webserver* yang dapat digunakan sebagai *web-bug* untuk merekam jejak aktifitas pengunjung web. Aplikasi dikembangkan menggunakan bahasa pemrograman *server side PHP*.

Tujuan dari penelitian ini adalah memanfaatkan *variable server* pada PHP yang digunakan sebagai *Web-bug* untuk merekam jejak aktifitas pengunjung web. Hasil penelitian ini tentunya dapat

dimanfaatkan untuk mengoptimalkan kemampuan aplikasi website yang dibuat agar mampu merekam aktifitas pengunjung sehingga memiliki data yang dapat dianalisa untuk mendukung pengambilan strategi pengembangan bisnis.

2. LANDASAN TEORI

2.1 Web Transaction Logs

Setiap *webserver* yang banyak digunakan saat ini memiliki fasilitas pencatatan transaksi antara *server* dan *client*, fasilitas ini dinamakan *Web transaction logs*. *Web transaction logs* (EPA, 1997) yang merupakan suatu sistem berada pada sisi *webserver* yang berfungsi untuk mengumpulkan informasi mengenai *client* dan aktifitasnya pada saat mengakses *webserver*. Informasi yang dikumpulkan dapat berupa:

- a. Waktu (Tanggal dan jam)
- b. Alamat Remote IP (*Client IP Address*)
- c. Nama pengguna (Jika ada otentikasi login)
- d. Method HTTP (GET / POST)
- e. Permintaan URL
- f. Kode Status HTTP
- g. Jumlah byte yang dikirim maupun yang diterima
- h. User agent / browser yang digunakan oleh client
- i. Informasi Cookie yang dikirimkan oleh client
- j. URL referensi
- k. Nilai parameter yang dikirimkan melalui URL (Method GET)

Pada *webserver* seperti Apache maupun IIS, terdapat berkas pencatatan transaksi *webserver* berupa berkas "log". Pada Apache menyimpan catatan transaksi data berupa berkas "access.log". Setiap *request* maupun transaksi data dipisahkan dengan baris baru.

2.1. Cookie

Cookie biasa juga dikenal dengan istilah HTTP *Cookie*, *Web Cookie* maupun juga *browser Cookie*. Apabila dilihat dari letak dan kegunaannya maka *cookie* dapat diartikan sebagai catatan yang digunakan oleh aplikasi website untuk mengirimkan informasi berupa status ke *webbrowser* pengunjung yang digunakan sebagai pengingat *webserver* terhadap pengunjungnya (Support Microsot, 2007). Pada dasarnya prinsip kerja dari komunikasi HTTP adalah sekali *request*, *response* dan selesai. Agar *webserver* tetap mengenali kliennya, maka diperlukanlah *cookie*. *Cookie* akan disimpan oleh *browser* dan akan dikirim ulang ketika *browser* melakukan *request* kembali kepada *webserver*.

Cookie bukanlah bentuk dari perangkat lunak, karena tidak dapat diprogram, membawa virus, maupun menjalankan file eksekusi (Support Microsot, 2007). *Cookie* dapat digunakan oleh *spyware* untuk mengumpulkan informasi mengenai aktifitas yang terjadi pada *browser*. Bahkan *cookie* dapat dicuri untuk mendapatkan akses pada layanan pemilik akun web. Kehadiran *cookie* pada *browser* tidak diketahui oleh penggunanya, karena penyimpanannya tidak ada notifikasi maupun konfirmasi. Dalam pengirimannya dari *Webserver* ke *browser*, *Cookie* berada satu bagian pada script *response* oleh *webserver*.

Jenis-jenis *Cookie* menurut dokumentasi Microsoft Support (2007), yaitu:

a. Session Cookie

Cookie dapat dimanfaatkan sebagai pencatat waktu sesi koneksi yang diperkenankan server kepada klien untuk mengakses suatu halaman website. *Session cookie* memiliki masa berlakunya, bila telah habis maka *Web-browser* akan menghapusnya.

- b. *Persistent Cookie*
Persistent Cookie adalah *cookie* yang disimpan untuk waktu yang lama dimana pengaturan waktu *Cookie* ditentukan oleh *webserver* saat mengirimkannya ke *browser*.
- c. *Secure Cookie*
Merupakan jenis *Cookie* yang dikirimkan melalui protokol HTTPS. *Secure Cookie* biasanya dimanfaatkan sebagai autentikasi, data kerahasiaan, Integritas (*Webserver* dapat mengetahui apabila *cookie* pernah dirubah), *Anti-Replay* (*Cookie* yang dicuri dapat dikenali oleh *webserver* sebagai *cookie* yang salah).
- d. *Third-Party Cookie*
Cookie yang dibuat oleh suatu alamat domain yang ditujukan kepada alamat domain lainnya, biasanya domain yang dituju adalah pihak yang menyelenggarakan iklan (pihak ketiga). Hal ini bertujuan untuk memberikan informasi yang dibutuhkan domain pihak ketiga.
- e. *Unsatisfactory Cookies*
Cookie yang digunakan untuk mengakses informasi pribadi yang dapat digunakan untuk tujuan sekunder tanpa adanya persetujuan ataupun pemberitahuan.

2.2 Web Bugs

Menurut Olsen (2011) mengenai *Web-bug* yang mendefinisikannya sebagai objek kecil yang ditanamkan pada halaman web maupun email untuk memberi tahu kepada suatu server berupa informasi kapan, siapa dan dimana mengenai suatu halaman web yang dibuka.

Pada umumnya *Web-bug* diimplementasikan berupa gambar kecil (1 *pixel* x 1 *pixel*). *Web-bug* merupakan salah satu cara untuk mengetahui siapa yang membaca suatu konten web

ataupun email. Hal ini untuk mengetahui apakah konten web disalin ke web lain, atau suatu email yang dikirimkan ulang ke alamat lain.

2.3 Database

DBMS merupakan bagian kritis pada aplikasi komputer modern saat ini yang banyak digunakan menurut Joseph(2007, 1). DBMS merupakan hasil penelitian dari beberapa tahun terakhir yang dikembangkan oleh akademi maupun dunia industri. Berdasarkan bentuk perangkat lunak yang tersedia, DBMS merupakan piranti lunak untuk membuat, mengelola serta mengakses basis data. Basis data dibentuk berdasarkan kolom dan baris (*record*), dan memiliki *attribute* baik berupa jenis data yang dapat disimpannya, kunci utama dan kunci tamu untuk membuat relasi. Banyak DBMS saat ini yang menggunakan bahasa standar SQL (*Structured Query Language*) untuk mengelola data yang ada di dalamnya.

3 METODE PENELITIAN

Metode penelitian yang digunakan adalah eksperimental, yaitu metode penelitian yang melakukan pengamatan pada aplikasi yang sudah ada dan dilakukan perubahan-perubahan *variable* yang disesuaikan dengan kebutuhan penelitian. Sedangkan metode pengembangan system menggunakan SDLC (*software development life cycle*) dengan model *Waterfall* dengan tahapan:

- a. Analisa
Pada tahapan ini penulis menganalisa fitur *web-bug* yang akan dikembangkan.
- b. Rancangan
Perancangan yang dibuat meliputi rancangan cara kerja aplikasi dan basis data untuk penyimpanan hasil *tracking*.

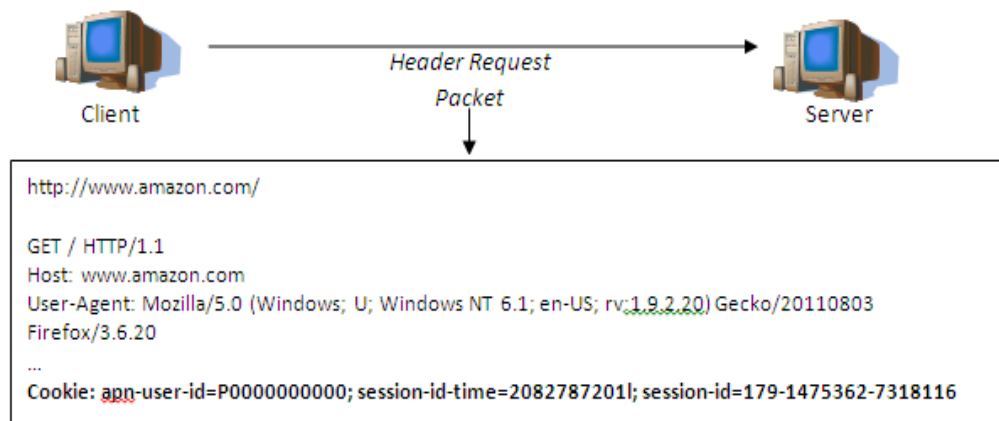
- c. Pengkodean
Setelah melakukan analisa kemudian diterapkan berupa kode program yang dibuat menggunakan PHP.
- d. Implementasi
Implementasi yang dilakukan yaitu dengan menempatkan *web-bug* pada laman web yang akan merekam aktifitas pengunjung web.

4 PEMBAHASAN

4.1 Analisa Transaksi Paket Data

Mengumpulkan informasi *client* berdasarkan aktifitas yang dilakukannya menggunakan teknologi web dapat dilakukan dengan memanfaatkan informasi HTTP *Request* yang dikirimkan oleh *browser*, serta catatan "*Cookie*" yang disimpan pada web browser, kemudian disimpan ke basis data yang didesain khusus untuk mengumpulkan informasi yang didapat dari *client*.

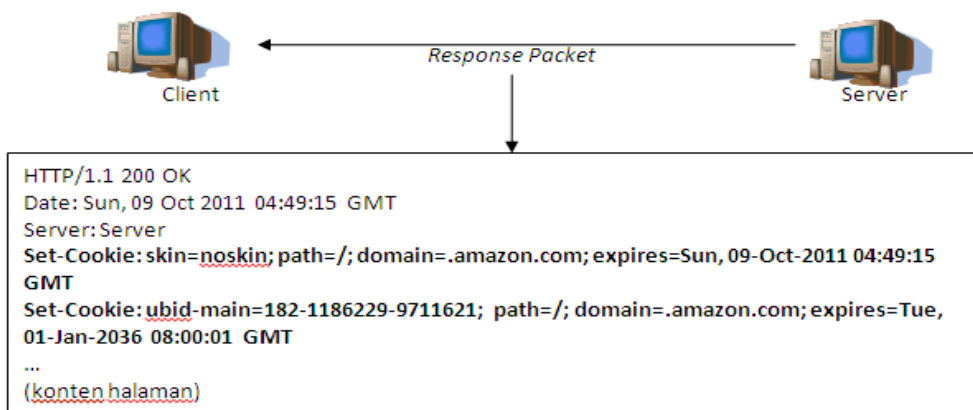
Berikut ilustrasi permintaan paket data dari *browser* ke *webserver*.



Gambar 1 . HTTP Request

Setelah *request* diterima oleh *webserver*, kemudian *server* akan melakukan pengolahan logika program serta menjalankan proses yang harus

dikerjakan, selanjutnya *webserver* mengirimkan *response packet* seperti berikut:

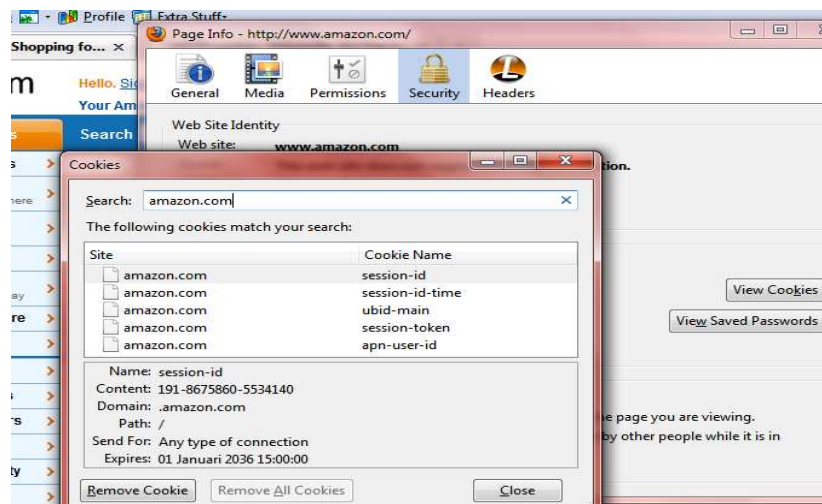


Gambar 2. HTTP Response

Pesan dari *header response packet* HTTP yang diterima oleh *client* berupa "Set-Cookie" akan membuat *browser* untuk menyimpan *cookie* tanpa disadari oleh penggunaanya. Namun bisa saja jika pengaturan pada *browser* diubah untuk tidak menyimpan *cookie* yang diberikan

oleh *Server*. Untuk mengetahui *Cookie* yang tersimpan pada *web-browser* pada Mozilla Firefox 3.6 bisa dilihat di:

Menu [Tools] -> Sub Menu [Page Info]->Tab [Security]->click Button [View Cookies]



Gambar 3. Lihat *Cookies* di Mozilla Firefox 3.6

4.2 Analisa Kemampuan *Web-Bug*

Sistem pencatatan yang dibuat untuk mengetahui aktifitas klien pada aplikasi web yang dibuat secara terintegrasi dan disimpan kedalam *database*, dapat dimanfaatkan untuk mengetahui:

- Siapa saja yang mengakses web
Berdasarkan informasi *cookie* maupun *session* dapat diketahui siapa yang mengakses aplikasi web. Apakah pengunjung biasa, atau pengunjung yang telah *login* (terdaftar).
- Dari mana saja asalnya
Informasi *IP address* dengan ketentuan wilayah IP dari masing-masing Negara, maka dapat diketahui dari Negara mana saja yang mengakses web. Dengan adanya data seperti ini, maka akan menjadi informasi yang sangat berguna untuk menentukan

produk sesuai target pasar yang lebih relevan.

- Halaman apa saja yang paling sering dikunjungi
Berdasarkan permintaan URL dari pengguna, maka dapat diketahui halaman yang paling banyak dilihat oleh pengunjung, sehingga dapat dijadikan informasi yang trend untuk disajikan kepada pengunjung lainnya.
- Transaksi data apa saja yang paling sering dilakukan
Transaksi data berupa permintaan halaman maupun pengiriman informasi dari *client* berdasarkan *form* yang disediakan pada aplikasi web. Contohnya pemesanan barang yang disimpan pada keranjang belanja. Contoh lainnya seperti Jenis produk apa yang dimintati oleh pengunjung dengan dibuatkannya *form polling* atau berdasarkan

kategori produk yang paling sering dicarinya.

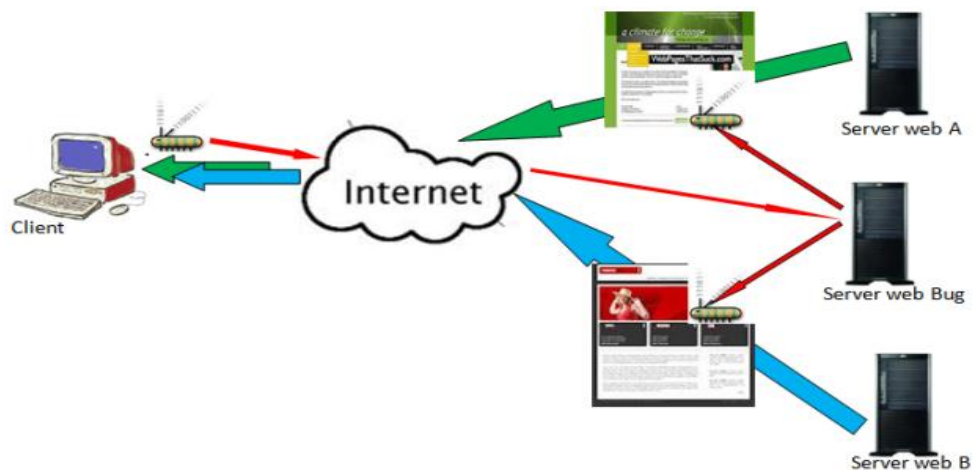
- e. *Browser* apa saja yang paling banyak digunakan
Berdasarkan informasi dari *User-Agent* maka dapat pula diketahui *browser* apa saja yang digunakan oleh pengunjung. Contohnya pada *browser* perangkat mobile, seperti *handphone* ataupun *gadget* memiliki informasi nama *user-agent* aplikasi perangkat mobile. Informasi mengenai pengguna mobile dapat menjadi pertimbangan untuk menyediakan layanan aplikasi web yang disesuaikan dengan tampilan *handphone* atau *gadget* agar lebih pas dilihat.

- f. Waktu kapan saja trafik tinggi (banyak pengunjungnya)

Dengan mencatat informasi waktu permintaan oleh *client*, maka dapat dijadikan *statistic* untuk menghitung jumlah permintaan terbanyak pada waktu yang berdekatan. Dengan informasi seperti itu, maka dapat menjadi pertimbangan untuk menentukan langkah kebijakan apa yang harus dilakukan untuk meningkatkan pelayanan.

4.3 Rancangan Cara Kerja Aplikasi

Berdasarkan cara kerjanya, penulis membuat ilustrasi alur web bug berkomunikasi sebagai berikut:



Gambar 4. Ilustrasi Pengiriman Web Bugs.

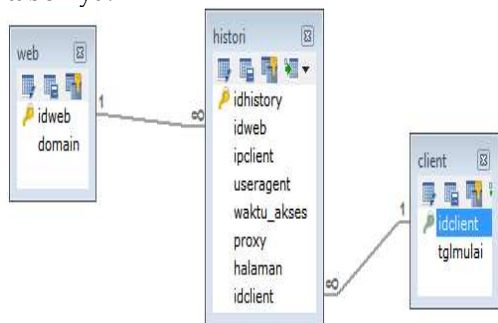
Dari gambar 4 dapat dijelaskan apabila seorang klien membuka website *www.A.com* yang mengandung *web-bugs*, maka *web-bugs* yang masuk akan mencatat *cookie* pada *browser* klien informasi mengenai ID *client* (jika *client* belum pernah dimasuki *web-bugs*), alamat website yang dibuka dan waktunya. Kemudian klien membuka website *www.B.com*, *web bugs* yang

dipanggil dari Web B juga mencatat mengenai alamat dan waktu mengakses halaman web B. *Cookie* akan selalu mengirimkan informasinya kepada nama domain yang sama, domain *web bugs* yang sama yang berada pada web A dan web B tentu akan mendapatkan segala informasi yang telah dicatat oleh *web-bug* di komputer client, sehingga *server web-bugs* akan memiliki *history*

mengenai urutan halaman, web apa yang sering dibuka oleh client berdasarkan id *cookie client* yang telah tersimpan.

4.4 Rancangan Basis Data

Agar informasi yang nantinya diterima oleh aplikasi web dapat disimpan, maka perlu adanya media penyimpanan yang nantinya dapat digunakan untuk mengolah data yang terkumpul menjadi informasi. Media penyimpanan yang dapat dimanfaatkan yaitu DBMS. Berikut rancangan tabelnya:



Gambar 5. Desain Database Histori aktifitas client.

Field	Type	Null	Key
idweb	int(4) unsigned	15B NO	PRI
domain	varchar(255)	12B YES	

Gambar 6. Tabel Web

Field	Type	Null	Key
idclient	int(8) unsigned	15B NO	PRI
tglmulai	datetime	8B YES	

Gambar 7. Table Client.

Field	Type	Null	Key
idhistory	bigint(8) unsigned	18B NO	PRI
idweb	int(4) unsigned	15B YES	MUL
ipclient	varchar(16)	11B YES	
useragent	varchar(255)	12B YES	
waktu_akses	datetime	8B YES	
proxy	varchar(255)	12B YES	
halaman	varchar(255)	12B YES	
idclient	int(8) unsigned	15B YES	MUL

Gambar 8. Tabel History

4.5 Pengkodean Program

Setelah hasil analisa dan rancangan yang telah dilakukan, langkah selanjutnya adalah melakukan pengkodean. Pada tahapan ini penulis melakukan eksplorasi terhadap perintah-perintah yang dapat dimanfaatkan pada pemrograman PHP yang digunakan.

Langkah eksplorasi kemampuan programing aplikasi web yang dilakukan yaitu melakukan *script* permintaan HTTP seperti yang ditampilkan pada Gambar 9. Melakukan permintaan kepada halaman www.amazon.com dengan melalui perangkat lunak browser Mozilla Firefox melalui referensi www.google.co.id. Dari informasi mengenai permintaan URL yang diterima *server*, maka *server* kemudian melakukan pengolahan data dan menanggapi dengan cara mengirimkan informasi balik kepada *client*.

Script HTTP Request dan *HTTP Response* itulah yang akan menentukan apa yang akan di-response oleh Server dan apa yang akan diterima oleh browser. HTTP Request yang dikirimkan oleh browser dapat dibaca oleh Server *Script Side*, sehingga pencatatan transaksi pada aplikasi web dapat dilakukan. Pada pemrograman PHP, Untuk mengumpulkan informasi mengenai aktifitas atau *request* yang dikirimkan oleh client dapat menggunakan perintah `$_SERVER`, `$_COOKIE`, `$_POST`, dan `$_GET`. Berikut contoh hasil penggunaan `$_SERVER` pada aplikasi web.

```

$_SERVER[HTTP_HOST] = diska.vacau.com
$_SERVER[HTTP_CONNECTION] = keep-alive
$_SERVER[HTTP_CACHE_CONTROL] = max-age=0
$_SERVER[HTTP_USER_AGENT] = Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.1 (KHTML, like Gecko)
Chrome/14.0.835.202 Safari/535.1
$_SERVER[HTTP_ACCEPT] = text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
$_SERVER[HTTP_ACCEPT_ENCODING] = gzip,deflate,sdch
$_SERVER[HTTP_ACCEPT_LANGUAGE] = en-US,en;q=0.8
$_SERVER[HTTP_ACCEPT_CHARSET] = ISO-8859-1,utf-8;q=0.7,*;q=0.3
$_SERVER[PATH] = /sbin:/bin:/usr/sbin:/usr/bin
$_SERVER[SERVER_SIGNATURE] =
$_SERVER[SERVER_SOFTWARE] = Apache
$_SERVER[SERVER_NAME] = diska.vacau.com
$_SERVER[SERVER_ADDR] = 31.170.161.196
$_SERVER[SERVER_PORT] = 80
$_SERVER[REMOTE_ADDR] = 222.124.156.242
$_SERVER[DOCUMENT_ROOT] = /usr/local/apache/htdocs
$_SERVER[SERVER_ADMIN] = admin@000webhost.com
$_SERVER[REMOTE_PORT] = 50395
$_SERVER[GATEWAY_INTERFACE] = CGI/1.1

```

Gambar 9. Hasil Pembacaan variable \$_SERVER pada PHP

Setelah diketahui *variable* apa saja yang ditangkap oleh *server*, kemudian langkah selanjutnya menggunakan *variable* yang ditangkap *webserver* untuk kemudian disimpan pada basis data

guna kepentingan lebih lanjut. Salah satunya sebagai data untuk dianalisa mengenai rekam jejak pengunjung. Berikut kode program yang dibuat menggunakan PHP.

```

<?php
require_once 'koneksi.php';
$idclient = isset($_COOKIE["idclient"])?$_COOKIE["idclient"]:"";
if($idclient=="")
{
    $idclient = mkIdClient();
    setcookie("idclient", $idclient,mktime()+(3600*24*7*4*12));
}
$httpx =
isset($_SERVER["HTTP_X_FORWARDED_FOR"])?$_SERVER["HTTP_X_FORWARDED_FOR"]
:"";
if($httpx!="")
{
    $ipclient = $httpx;
    $ipproxy = $_SERVER["REMOTE_ADDR"];
}else{
    $ipclient = $_SERVER["REMOTE_ADDR"];
    $ipproxy = "-";
}
$useragent = addslashes($_SERVER["HTTP_USER_AGENT"]);
$webid = (int) (isset($_GET["webid"])?$_GET["webid"]:"");
$halaman = addslashes(isset($_GET["p"])?$_GET["p"]:"");

$query = "INSERT INTO histori(idweb, ipclient, useragent, waktu_akses,
proxy,
        halaman, idclient)
VALUES($webid, '$ipclient', '$useragent', NOW(), '$ipproxy',
        '$halaman', $idclient)";
@mysql_query($query);?>

```

Gambar 10. Potongan kode program *web-bug*

Cara kerja dari kode program yang dibuat yaitu:

- Baca Cookie "idclient" dari client
- Jika Cookie idclient tidak ada, maka buat idclient dan kirimkan cookie berupa idclient lama waktu setahun
- Baca IP forward HTTP
- Jika IP Forward HTTP ada maka IP proxy adalah alamat IP Remote, IP client sebenarnya adalah IP Forward, jika tidak ada maka IP proxy kosong dan alamat IP client adalah IP Remote.
- Baca user agent, web id (no id web yang memasang aplikasi ini), halaman yang dicatat
- Simpan ke table histori.

Penggunaan kode di atas harus diikutsertakan pada setiap halaman website yang dibuat. Maka dengan adanya pencatatan seperti ini, aktifitas pergerakan pengunjung akan selalu dicatat dan menjadi data yang berguna untuk mengetahui pola kecenderungan ketertarikan pengunjung terhadap produk yang ada ditawarkan di web.

Informasi berharga dari kumpulan data yang tersimpan di basis data dapat ditemukan apabila dikelola dengan memanfaatkan datawarehouse maupun data mining untuk mendapatkan informasi pola aktifitas pengunjung.

4.6 Implementasi Web-bug

Untuk menerapkan web-bug yang telah dibuat caranya dengan menyisipkan kode HTML *image* (img) atau dapat pula menggunakan elemen *frame* (iframe) yang disertakan *style display none* untuk menyembunyikan elemen yang diletakan pada bagian dokumen HTML. Contoh penerapan sebagai berikut:

- Penerapan kode penerapan web-bug menggunakan *image*

```

```

- Penerapan menggunakan *frame*

```
<iframe  
src="http://alamatserver/webbug.p  
hp?p=halweb.com"  
style="display:none"></iframe>
```

5. KESIMPULAN

Aplikasi yang dibahas pada tulisan ini mengenalkan beberapa teknologi yang sering digunakan, seperti *web transaction logs* sebagai cara untuk mengumpulkan informasi transaksi antar *webserver* dan *client*. *Web-bugs* sebagai cara yang digunakan untuk mengumpulkan informasi aktifitas aktifitas pengguna di internet dari satu halaman web ke halaman web lain. Teknologi *website* melalui protocol HTTP(s), pada dasarnya tidak dapat mengingat adanya komunikasi antara *client* dan *server* apabila tidak ada suatu catatan yang diberikan baik dari *server* maupun dari *client*. *Cookie* menjadi bentuk catatan pengingat komunikasi antara *server* dan klien. Eksploitasi kegunaan *cookie* yang tidak lagi sebagai pengingat komunikasi *webserver* dan *browser*, saat ini dapat menjadi catatan untuk memonitoring apa yang dilakukan oleh pengguna di internet.

Keamanan informasi pribadi seseorang merupakan suatu permasalahan yang kadang tidak diperhatikan oleh para marketer maupun organisasi di dunia maya. Dengan maksud untuk mendapatkan informasi pasar yang tepat agar perdagangan *online* menjadi lebih tepat sasaran dengan tujuan meningkatkan penjualan suatu produk, pengumpulan informasi pengguna serta aktifitas yang dilakukannya menjadi kumpulan data yang dapat diperjualbelikan kepada pengiklan atau organisasi yang membutuhkannya. Pada akhirnya

mekanisme pelacakan aktifitas maupun pengumpulan informasi mengenai pengguna semakin beragam pada saat ini. Privasi kegiatan seseorang di internet bukan lagi hal yang sifatnya rahasia.

DAFTAR PUSTAKA

- Farmer et al. 2002. *A Brief History of Modern RDBMS IT Management (Article 3)*. <http://www.mountainman.com.au/software/history/it3.html> (11 May 2014)
- Joseph et al. 2007. *Architecture of a database system*. <http://db.cs.berkeley.edu/papers/fntdb07-architecture.pdf> (11 May 2014)
- Kiang et al. 2000. Marketing on the Internet — who can benefit from an online marketing approach? http://faculty.fuqua.duke.edu/~khs hang/Working%20Papers/Marketing_on_the_Internet.pdf (11 May 2014)
- Liu et al. 2005. *A Secure Cookie Protocol*. IEEE XPlore. <http://www.cse.msu.edu/~alexliu/publications/Cookie/cookie.pdf> (11 May 2014)
- Meyer, Rugor. 2008. *Detecting Attacks on Web Applications From Log Files*. <http://www.testexpert.com.br/files/Detecting%20Attacks%20on%20Web%20Applications%20from%20Log%20Files.pdf>. (11 May 2014)
- Microsoft Support. 2007. Description of Persistent and per-Session Cookies in Internet Explorer. <http://support.microsoft.com/kb/223799/EN-US> (11 May 2014)
- Microsoft Windows XP Professional Product Documentation. Understanding Cookies. http://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sec_cook.msp?mfr=true (10 May 2014)
- Olsen, Stefanie. 2000. *Nearly Undetectable Tracking Device Raises Concern*. <http://news.cnet.com/2100-1017-243077.html> (10 May 2014)
- Scott Hamilton, Helen Thurlow. 2005. *Transaction Log Analysis @ State Library of Queensland*.